

How Students Construct Human–AI Boundaries in Software Engineering Teamwork

Isabella Graßl  

Technical University of Darmstadt, Germany

Abstract

Background. Software engineering is a sociotechnical discipline in which teamwork is central, even as AI tools are integrated into professional and educational practice. While prior research has primarily examined AI as an individual productivity tool, we know little about how software teams position AI within collaboration.

Aims. Drawing on boundary work theory, we explore how students integrate AI into project-based software engineering teamwork and how they construct boundaries between AI and human collaboration.

Method. We conducted a qualitative multiple-case study across seven software engineering courses, collecting interviews ($n = 27$) and reflections ($n = 89$) from 116 students.

Results. Students construct explicit human–AI boundaries within software engineering teamwork. They draw *complementary* boundaries, using AI to support individual preparation and thereby improving the conditions *around* collaboration, and *competitive* boundaries, actively excluding AI from interpersonal aspects of teamwork that they reserve for humans, grounded in concerns about accountability, trust, and relational value. We found no evidence in students’ accounts that they perceived AI as a *co-evolving* collaborative partner.

Conclusions. We argue that students actively construct AI as infrastructural support rather than as a social collaborator, thereby preserving human responsibility, empathy, and trust within software engineering teamwork. These findings contribute to emerging discussions on human–AI collaboration and the evolving role of software engineers in this era.

2012 ACM Subject Classification Software and its engineering → Collaboration in software development; Applied computing → Education

Keywords and phrases Software Engineering, AI, Teamwork, Collaboration

Digital Object Identifier 10.4230/LIPICs.ESEM.2026.31

Category Technical Track Paper

1 Introduction

Software engineering is a collaborative discipline [37] that requires communication, coordination, negotiation, and shared responsibility between team members [25, 39]. Yet, collaboration is not easy to learn: professional and student teams alike often face unequal participation, communication breakdowns, and interpersonal conflict [20, 12]. In the last few years, professional and educational software engineering [47, 42] has been adapting to AI tools such as GitHub Copilot and ChatGPT [40]. They support (future) developers with coding and documentation, often boosting perceived individual productivity [23, 32].

Most software engineering research has treated AI as an individual tool, leaving its role in teamwork largely unexplored. Studies in social sciences and HCI suggest that AI can influence team communication, commitment, and perceived productivity [31, 15, 2], but these insights have not been systematically examined in software engineering, where interpersonal skills are often undervalued [35, 24], and students must learn to collaborate, share responsibility, and build trust [18]. This is especially important since students are not only learning technical



© Isabella Graßl;

licensed under Creative Commons License CC-BY 4.0

20th International Symposium on Empirical Software Engineering and Measurement (ESEM 2026).

Editors: Robert Feldt, Maria Paasivaara, Daniel Mendez, Stefan Wagner, and Marvin Muñoz Barón;

Article No. 31; pp. 31:1–31:21



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

practices, but are also developing professional norms and expectations regarding collaboration, responsibility, and the role of AI in software engineering work. Understanding how students integrate AI into teamwork may therefore provide insight into broader sociotechnical shifts emerging in software engineering practice.

To investigate this, we draw on boundary work theory [21], which examines how people define and maintain distinctions between domains of responsibility and legitimacy. We draw on three forms of human–AI boundary work from recent work [17]: *complementary boundaries*, where AI and humans take on distinct but supporting roles; *competitive boundaries*, where humans actively resist AI involvement in domains they consider their own; and *co-evolving boundaries*, where AI is treated as a partner that develops together with humans over time [17]. This perspective enables us to explore how they position AI in relation to human collaboration [39, 32].

By analysing interviews and written reflections from 116 students from project-based software engineering courses, we investigate how students use AI in project-based teamwork (RQ1), what functional and practical changes emerge (RQ2), and how they conceptually position AI in relation to humans (RQ3):

RQ1: How do students integrate AI tools into their software engineering teamwork?

RQ2: How do students perceive the practical and functional impact of AI tools on collaboration within their software engineering teamwork?

RQ3: How do students normatively position AI tools in relation to their human teammates in software engineering teamwork?

Our findings show that students draw a consistent line between AI supporting the conditions *around* collaboration and AI participating *within* it. They position AI as support for individual preparation and productivity rather than as part of their collaboration itself. While AI improved conditions around teamwork, students actively reserved interpersonal, emotional, and accountable aspects of collaboration for humans. According to our data, students did not perceive AI as an evolving collaborative partner within their teamwork. Instead, they maintained clear human responsibility within their teamwork, even when relying on AI tools for technical support. We argue that the adoption of AI tools depends not only on technical skills but also on *values* about what work should remain human, emphasising that while tools evolve, the social character of software teamwork is still a central aspect.

2 Background and Related Work

Software engineering is a sociotechnical discipline in which communication, coordination, and shared responsibility are central to software development [37]. Thus, software engineering must prepare its students not only for technical implementation but also for teamwork [37]. However, teamwork in software engineering courses is often challenging [20]. Prior work reports recurring issues including unequal participation, ineffective communication, coordination difficulties, poor planning, and interpersonal conflict [18, 30, 26, 12]. These challenges disproportionately affect underrepresented groups, including female and international students [12]. Technologies can support coordination, but may also reinforce inequalities when access, confidence, or digital literacy differ [27]. With the presence of AI, this challenge becomes even more urgent [29], as software systems must now address fairness, transparency, diversity, and accountability alongside traditional quality goals [3].

2.1 AI in Software Engineering (Teamwork)

Generative AI tools such as ChatGPT and GitHub Copilot are increasingly integrated into software engineering education and practice [19, 29]. Prior research has primarily examined AI as an individual productivity tool, focusing on coding support, debugging, assessment, and learning outcomes [13, 32, 29]. Studies suggest that students perceive AI as useful for idea generation, programming support, and productivity gains, while also expressing concerns regarding overreliance, responsibility, and reduced critical thinking [38, 13].

More recently, research has begun to examine AI within collaborations. Prior studies in other disciplines suggest that AI can shape communication patterns, collective attention, and team interaction [15, 49]. Trust and the alignment of shared goals in a task play a key role since misunderstandings with AI reduce trust, communication, and perceived performance [9, 2, 36]. Human–AI teams may show lower commitment and motivation than all-human teams, weakening collaboration despite technical assistance [45].

In software engineering specifically, a recent mapping study finds that human–machine teaming remains rare in software engineering practice [33], and position papers have begun to sketch AI as a future collaborator in student software teams [16]. At the individual level, developers describe AI as a subordinate helper, a junior colleague or teacher [48], and trust in tools such as Copilot shifts over the course of individual use [36].

However, developers in pair programming treat AI contributions differently from human input, reacting less critically, which may shift norms of responsibility and judgement in collaboration [44]. Despite growing interest in human-AI collaboration, empirical understanding of how software engineering teams socially position AI within teamwork remains limited.

2.2 Boundary Work in Human-AI Collaboration

Abrahão et al. [1] project AI as a future teammate in software engineering, which raises the question of whether students already treat AI this way, or whether a tool-based view remains the norm. To examine how our future software engineers position AI within teamwork, we draw on boundary work theory [21]. Boundary work describes how actors define and maintain distinctions between domains of responsibility, legitimacy, and practice. In the context of generative AI, Kim et al. [17] describe three forms of human-AI boundary work. *Complementary boundaries* position AI and human work as distinct but mutually supportive, where AI assists selected tasks while humans retain responsibility for others. *Competitive boundaries* emerge when actors resist AI participation in domains perceived as fundamentally human or socially important. *Co-evolving boundaries* arise when humans and AI are treated as developing collaborative partners over time. These three boundary types are not mutually exclusive and may operate simultaneously across different domains of practice. This framework enables us to move beyond describing AI use towards analysing how students construct (un)acceptable forms of AI participation in software engineering teamwork. This matters for software engineering education, where the question is not only what AI can or cannot do, but what students believe it should and should not do, and why.

3 Methods

The aim of this study is to understand how students use AI tools in project-based software engineering courses and how they perceive the role of AI in their collaboration. We used a qualitative multiple case study design [46], which allowed us to systematically explore the practices in educational settings by combining multiple data sources [46]. This design also enabled us to compare across contexts, hence different universities and course settings with the same overall aim and structure.

3.1 Cases: Course Contexts

We studied seven project-based courses in software engineering across three public universities: one lab course at the University of Victoria in Canada (*North America, NA*), two lab courses at the University of Passau in Germany (*Europe, EU1*), and two lab courses plus two seminar courses at the Technical University of Darmstadt in Germany (*Europe, EU2*). Each course represents one case ($n = 7$ cases). All courses followed a project-based learning approach, which connects learning to real-world issues and makes tasks more meaningful. This approach includes improving students' agency, meaning they *own* their projects from topic selection to presentation. They began with theoretical input at the beginning of the term, and then worked in teams to apply research methods and software engineering principles to their projects. Across courses, students produced outputs such as papers, reports, proposals, peer reviews, and posters.

All courses required students to work in teams over one academic term (four to five months), and to produce concrete outcomes. Team sizes ranged from four to six students in all courses with the same credit points, which makes these courses suitable cases for studying AI use in teamwork. All courses focused on human-centred software development, where user needs and human-focused research play a central role. The main difference between formats lies in the focus: labs emphasised implementation, while seminars emphasised conducting research as a practical contribution.

Human-centred software development lab. Students developed a software solution for a real or simulated client, implemented a prototype, and wrote a short report. While technical implementation was important (e.g. architecture and testing), user needs formed the core of the project. The course followed design thinking principles. Teams also worked with agile practices: they organised their own schedules, task allocation, and feature planning. In the final report, students explained how they addressed diverse user needs and justified their technical design and testing decisions.

Human-centred software development seminar. Student teams developed a research idea in software engineering, implemented a prototype, and wrote a research report. Students received an initial set of research papers in human-centred software engineering, which served as starting points for topic development and discussion. The seminar placed stronger emphasis on reflection, academic writing, and critical discussion than the lab format.

All courses were elective and open to both Bachelor and Master students. The mix of Bachelor and Master students in the same courses meant participants had varying levels of programming and teamwork experience as well as professional and personal maturity. However, the focus on human-centred software development created common ground. We found no systematic differences in AI collaboration perceptions between Bachelor and Master students in our analysis. The first author taught all courses except the lab in *NA*, where she was not the formal course instructor but co-supervised the student teams.

All three universities at the time had adopted standard institutional policies on AI use, similar to ACM guidelines: students must acknowledge AI use and avoid copy-pasting without attribution. Within courses, we applied no additional restrictions. Students were informed that any directly copied AI-generated code or text must be clearly marked as such, but they remained free to use AI as they wish for their learning and project work. This approach allowed us to observe their authentic AI use patterns rather than keeping strict rules.

3.2 Data Collection and Participants

We used two complementary data sources: written reflections and semi-structured interviews. This combination supports data triangulation and strengthens the credibility of our study [28].

3.2.1 Interviews and Reflections

The written reflections and interview guide followed the same core questions. Interviews allowed us to get further insights, ask for clarification, and concrete examples. We designed the interviews following established guidelines [34].

1. **Background Information (optional).** Gender, study program.
2. **Practical AI Use in Teamwork (maps to RQ1).** *For which activities have you used AI tools in this project and which tools did you use? Please elaborate on whether they were helpful or not for your team and provide examples.*
3. **Perceived Impact of AI on Teamwork (maps to RQ2).** *How has using AI tools affected your collaboration with your team partners, if at all? Please provide examples.*
4. **Comparison Between Human and AI Collaboration (maps to RQ3).** *How did you experience collaborating with AI compared to collaborating with your team partners?*
5. **Broader Teamwork Experiences (context for all RQs).** *Reflect on your team dynamics in this course. What has been challenging? What has been easy?*

The same mappings of the research questions apply to the reflection instrument. We transcribed all interviews using *Otter.ai* and *Notta.ai*, and manually checked and corrected the transcripts. Written reflections were already available in textual form. We imported all transcribed data into *Atlas.ti* for the reflexive thematic analysis.

3.2.2 Recruitment

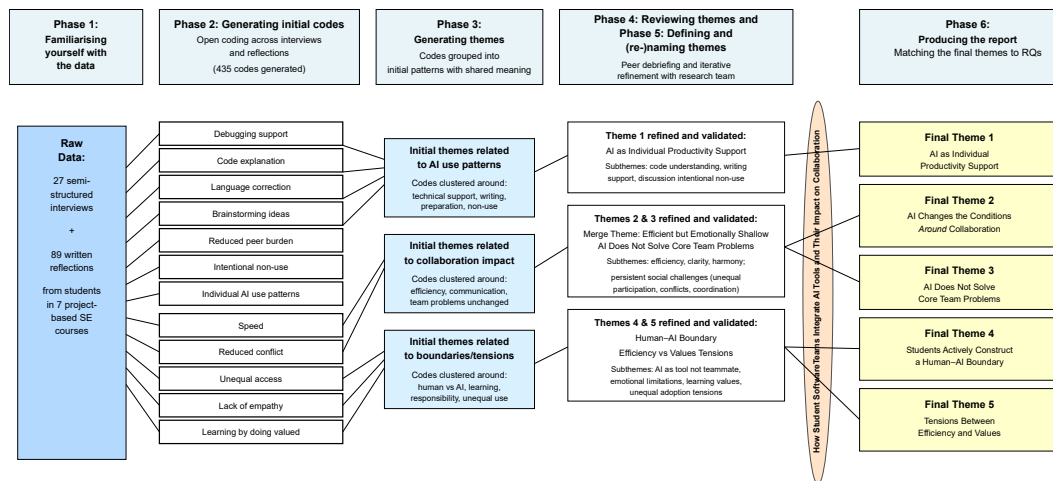
We invited all students enrolled in the selected courses to participate. Participation in reflections and interviews was voluntary, took place at the end of each course, and had no influence on grading, as all grades had already been submitted. We distributed the link to the online reflection form and the invitation to participate in interviews at the end of each course, coinciding with the end of term. Students completed the reflections online.

We conducted interviews only in the NA and EU1 contexts due to practical constraints; data from EU2 consist entirely of written reflections. This approach ensured that all participants could contribute in a format appropriate to their context. Data collection took place between January 2024 and December 2025. All interviews were conducted in person and audio-recorded with participants' informed consent.

For reflections, we achieved a 46% response rate (89 responses from 193 enrolled students). For interviews, all invited students participated (27 of 27). We collected 89 individual written reflections across four courses in *EU2*. The completion time for reflections ranged from 8 to 18 minutes. We conducted 27 semi-structured interviews, with durations ranging from 21 to 43 minutes. Of these participants, 16 were from *EU1* and 11 from *NA*.

3.2.3 Participants

Our dataset includes perceptions from 116 students. 31% of students identified as female and 69% as male; no participants identified as non-binary. Most participants studied Computer Science (83%), with the remaining in Business Informatics. The majority were undergraduate students (74%, Bachelor, 26% Master). We did not systematically capture students' prior



■ **Figure 1** Overview of the reflexive thematic analysis process from raw data to final themes. The figure illustrates how data from 116 participants were analysed through the six phases of reflexive thematic analysis [4], resulting in five themes.

experience with, frequency of use of, or confidence in generative AI tools before the course; all participants stated that they are familiar with AI tools and have already used them within their studies. We obtained ethics approval from the committee for interviews in Canada. In the German context, formal approval was not required; however, we followed standard ethical guidelines in both settings, including informed consent, voluntary participation, and anonymisation. We obtained consent from all participants. All data were stored securely using *GitLab* on German servers. We replaced all participant identifiers with pseudonyms (e.g., NA-01, EU1-02) during transcription.

3.3 Data Analysis

We analysed the data using reflexive thematic analysis, following Braun and Clarke’s established approach [7]. This approach conceptualises the coding process as an interpretative, researcher-positioned practice [6] and aims to capture the range of meanings present in the data, rather than to produce a definitive model [34]. Consistent with Braun and Clarke’s philosophy, we emphasise that qualitative research treats “*researcher subjectivity as a resource rather than a threat to knowledge production*” [5]. In order to verify our analytical process, we followed best practices for qualitative software engineering [22] and documented the COREQ [41] checklist in our replication package.

3.3.1 Reflexive Thematic Analysis

Figure 1 provides an overview of the process. One researcher (the corresponding author) conducted the initial open coding and theme generation across the full dataset (Phases 1–3); a second researcher (research assistant) acted as peer debriefer in Phases 4–5, reviewing the coding and thematic structure and challenging interpretations, without independently coding the raw data. Interviews and reflections were coded within the same coding frame in *Atlas.ti* and treated as a single combined dataset per case; we compared emerging patterns across the two sources during Phase 4 rather than analysing them as separate strands.

In the first phase of reflexive thematic analysis, we familiarised ourselves with the data by reading all transcripts *actively* [4].

In the second phase, one researcher conducted open coding across the entire dataset. We adopted an inductive approach, meaning that codes were derived directly from the data rather than from predefined categories [4]. The coding combined semantic and latent approaches [5]. This involved line-by-line coding with close attention to participants' wording. In line with Braun and Clarke's guidance [7], we kept initial codes close to the original expressions used by students. This process resulted in 435 initial codes.

In the third phase, we iteratively refined codes and grouped them into higher-level categories, thereby *generating* initial themes. As Braun and Clarke emphasise, themes are not passively located *in* the data but are actively constructed by researchers through interpretation and abstraction [5]. Accordingly, our themes do not represent topic summaries but patterns of *shared meaning* across the dataset [5]. This phase also involved repeated comparison across participants and data sources, resulting in seven initial themes.

In the fourth phase, we refined the themes by identifying the *essence* of each theme: clarifying what *story* each theme tells and how it is distinct from the others. We used peer debriefing to strengthen rigour [14]. A research assistant reviewed the coding and thematic structure, challenged interpretations (e.g., by questioning overlaps between themes), and discussed alternative readings [5]. This process led to further refinement: we merged one theme with another and renamed two themes. The full analytic trail, including refinements and revisions, is documented in our replication package. During this phase, we also finalised sub-themes, which provide internal structure and nuance within each theme [4]. We compared themes across both data sources (interviews and reflections) and found consistent patterns, which strengthened confidence in the findings. Interviews provided richer contextual detail, while reflections offered broader coverage across participants.

Finally, we structured the themes in relation to the three research questions. Themes addressing how and for what purposes students use AI tools correspond to RQ1 (*Theme 1*, Figure 1). Themes addressing perceived effects on communication and collaboration correspond to RQ2 (*Theme 2*, *Theme 3*). Themes addressing how students distinguish between AI tools and human collaborators correspond to RQ3 (*Theme 4*, *Theme 5*).

Consistent with reflexive thematic analysis, we did not aim for interrater reliability, as coding is understood as an interpretative act shaped by researchers' perspectives rather than an objective process to be standardised [7]. Instead, we ensured rigour through peer debriefing, critical discussion, and sustained reflexive engagement throughout the analytic process [6]. Following Braun and Clarke's guidance [7], we do not report or interpret these counts, but they are visible in our replication package through *Atlas.ti*.

We selected illustrative quotes from all available and consented data. One researcher compiled an initial set of candidate quotes, and the research assistant reviewed these selections. In cases of disagreement, both researchers discussed the selection. The first researcher typically deferred to the second researcher's judgement, as the latter had greater analytic distance from the data collection process.

3.4 Trustworthiness

Consistent with interpretive qualitative principles of reflexive thematic analysis, we evaluate the quality of our study using the framework of trustworthiness.

Credibility and Confirmability. The corresponding author, who conducted all interviews and led the coding, identifies as a white European woman, is a first-generation academic, and has several years of experience in software engineering research and education. Her attitude

towards AI tools can be described as cautiously sceptical yet open. This positioning may have influenced how the data were interpreted and presented. Through ongoing discussions within the research team, where some colleagues hold very positive views of AI and are frequent users of AI tools, we aimed to balance this influence and critically reflect on our own assumptions. We also acknowledge that the first author’s dual role as teacher and researcher may have influenced student responses. However, we collected data only after final grades were submitted, and participants expressed diverse and sometimes critical perspectives on AI use, suggesting they felt able to share their views openly.

We triangulated two data sources (reflections and interviews) and examined whether themes appeared consistently across both. We kept codes close to participants’ wording to remain grounded in the data. Some question phrasing may still have shaped responses; we mitigated this risk by basing one question on prior work and by jointly discussing and refining all questions within the research team. Participation in reflections and interviews was voluntary, which may introduce self-selection bias, particularly given the human-centred focus of the lab and seminar. The study relies on self-reported data; participants may not accurately recall their AI use or may describe their behaviour in socially desirable ways. Qualitative analysis also involves interpretation. To reduce the risk that findings reflect only one researcher’s perspective, we used iterative coding, peer debriefing, and critical discussion within the research team. For instance, during peer debriefing, the research assistant questioned the initial separation of what became Theme 2 and Theme 3, arguing that *efficiency gains* and *unresolved team problems* described the same underlying tension from two angles; this discussion led us to merge the two initial themes and reframe them as complementary outcomes of the same boundary.

Transferability. We aim to offer theoretically meaningful insights rather than representativeness. We provide detailed descriptions of the context, participants, and themes to support readers in judging transferability to their own settings. We note that our data capture individual students’ accounts of AI use within team contexts; claims about team-level dynamics are inferential. Including multiple courses across different universities and countries strengthens relevance beyond a single local context. At the same time, the findings represent a snapshot from January 2024 to December 2025; students’ practices and perceptions may evolve as tools develop and educational norms around AI use continue to stabilise. All courses shared a human-centred pedagogical orientation; this may have primed students to value human interaction and could sharpen the boundaries reported here relative to other software engineering course contexts. In addition, since participation, though voluntary, took place within an academic setting, some competitive boundaries may partly reflect institutional norms around originality and independent learning rather than AI’s capabilities as such, as illustrated by students’ concerns about plagiarising and learning progress. Practitioners may draw these boundaries differently; we return to this as a direction for future work.

Dependability. We documented and provide the full analytical circle, including the generation of codes and themes, as well as the COREQ checklist.

4 Results

We developed five themes across interviews and written reflections (Table 1). All themes appeared across all seven courses; the main variation was in emphasis, as lab courses more often discussed AI in relation to coding, and seminar courses in relation to writing and idea development. We provide a cross-case comparison table in the replication package.

■ **Table 1** Overview of themes from the reflexive thematic analysis.

Theme	Subthemes	Codes (examples)
1: AI as Individual Productivity Support	AI to understand and debug code AI to improve written communication AI to prepare for brainstorming and discussions Selective and critical AI use	debugging support; explanation of code; refactoring language correction; documentation; post-processing brainstorming; idea generation; pre-meeting preparation trust concerns; low-quality output; team contracts
2: AI Changes the Conditions Around Collaboration	Experiencing efficiency gains and availability Using AI for clarity and communication Valuing AI's harmony and predictability	time savings; speed; AI availability; reliability language barrier support; reduced peer burden emotionally stable; low-friction; less conflict; agreeable
3: AI Does Not Solve Core Team Problems	Communication issues Unequal workload distribution Scheduling and coordination Interpersonal conflict and feedback Diversity-related conflict	not speaking up; difficulty motivating; defensive reactions unequal participation; compensation for others conflicting schedules; different priorities; job scheduling criticism as personal; defensiveness; unresolved tensions not taken seriously; disrespect; lack of integration
4: Students Actively Construct a Human-AI Boundary	Treating AI as tool, not teammate Recognising AI's emotional limitations Reserving empathy and support for humans Human responsibility and accountability	cognitive offloading; distrust; one-sided relationship missing support; lack of empathy; trauma bonding categorical rejection for interpersonal problems human judgement; trust in humans; overreliance
5: Tensions Between Efficiency and Values	Tensions from unequal AI adoption Prioritising human learning over efficiency Emotions and purpose as human values	authenticity of work; uniformity; unequal access learning by doing; value of effort and struggle; dependency meaningful discussion; shared emotional experience

4.1 RQ1: AI Usage in Software Teams

Students described AI as functioning as preparatory infrastructure that strengthens individual readiness before collaboration. They view AI tools as support for their individual work rather than a shared team resource (*Theme 1*, Table 1). They explained AI as helping them *individually* to brainstorm, code, document, organise, and prepare for meetings, but rarely as something that is directly used in their collective work.

4.1.1 Theme 1: AI as Individual Productivity Support

AI as preparatory infrastructure. Students used AI to explain unfamiliar code, debug errors, refactor implementations, and support technical understanding. Rather than directly copying outputs, they described AI as helping them reason through problems independently to *understand why using exactly this code* (NA-12). This explanatory function became particularly important when teams worked with unfamiliar technologies:

“Thank God for ChatGPT. As none of us has used the framework Flutter or the language Dart, we needed this app coded basically in a month. We got to learn now really quick and apply it. Flutter is really new and doesn't have helpful documentation.” – NA-10

Students also used AI for writing support, including documentation, grammar correction, and report writing. They explained that they do not have to worry about *describing code details or structuring formal sentences* (EU2-50), since those are tasks they are bored with or

31:10 How Students Construct Human–AI Boundaries in Software Engineering Teamwork

find challenging. Support for communication was particularly relevant for students working in a second language. However, even when collaborative artefacts were produced, AI use remained individualised since work was divided between team members.

Beyond implementation and writing, students described AI as useful for brainstorming and meeting preparation. AI-supported preparation enabled students to arrive at discussions with more structured ideas and clearer proposals. This preparation had an indirect positive effect on team meetings since, instead of arriving at discussions unprepared, they used AI to develop outlines, ideas, and options beforehand:

“I used AI to help brainstorm in preparation for our team discussions and standups. Before a project meeting, I gathered a list of possible approaches from AI chats and shared them with the team. Having those options ready meant we could start the conversation with a clear direction and immediately, get everyone involved faster, and focus on improving the ideas together, which was positive.” – EU2-16

Selective and critical AI use. Students did not adopt AI uncritically as they described careful, selective, or even deliberate non-use of AI. They emphasised the need to edit AI output, verify correctness, and apply their own judgement. Some teams introduced explicit rules in their team contracts at the beginning of the term, while others avoided AI for personal or ethical reasons. They framed teamwork itself as inherently human:

“We did not use AI tools, and it did not occur to me to use it once. Team work relies on human participation, so how could those help us?” – EU2-30

Students with professional experience were particularly sceptical about overreliance on AI-generated code, using it rather as *last resort* (EU1-02), and preferred human-validated resources such as Stack Overflow as they valued the collective human knowledge:

“ChatGPT just gives you code, but you don’t know where to put it or how it works. You need to ask more and more questions. I don’t wanna go down this rabbit hole, it’s just annoying.” – EU1-11

RQ1. Students use AI not *within* the team, but *individually* to strengthen personal preparation and productivity before contributing to collaborations.

4.2 RQ2: Impact of AI on Collaboration

While students reported several positive impacts, these related to efficiency, preparation, and clarity, improving the conditions for collaboration without transforming the collaborative process itself; hence, AI influenced collaboration indirectly (*Theme 2*, Table 1). Some core team problems remained unaddressed despite the use of AI tools (*Theme 3*).

4.2.1 Theme 2: AI Changes the Conditions *Around* Collaboration

Reducing coordination and communication friction. Students valued AI for its speed, availability, and predictability. AI enabled them to access support immediately without depending on teammates’ schedules or availability, as with *AI it is easier, because it is always available when you ask for it.* (EU2-43). This saved them time they would otherwise have spent arranging a meeting to *just clarify a single line of code* (EU1-10), whereas *AI tools allow for faster feedback cycles* (EU2-16). Beyond scheduling, students also valued that AI has no mental or emotional limitations:

“AI tools are always available, not only scheduling wise but also mentally. There is no bad day for them so you have to work around that or postpone mentally taxing work to accomodate for e.g. a bad nights sleep.” – EU2-08

In this context, students contrasted AI interaction with human collaboration, describing AI as emotionally neutral and less socially demanding:

“With an LLM there is no conflict to resolve, as they follow what you tell it to do. That way we could focus on implementation rather than on small things.” – EU2-22

In its enhanced form, students used AI to soften emotionally charged communication before interacting with teammates:

“I also used it when I was once again absolutely disappointed by the team, and then typed a very aggressive message into Discord and thought to myself, I can’t send this like this. Then I put it into ChatGPT and said, “Please rewrite this in a factual and unemotional way, or maybe a little friendlier.” Only then did I send it.” – EU1-03

Students also reported that AI supported clearer communication, particularly for international students and those with lower confidence in English. In these cases, AI reduced some of the communicative barriers surrounding teamwork:

“As a foreigner, I believe that, due to the language barrier, using AI tools has made communication more efficient by helping me understand things I couldn’t or correcting mistakes in my speech, and looking up unfamiliar words, so I did not feel so awkward around the team.” – EU2-04

Some of these effects operated primarily at the individual level: reduced awkwardness or increased confidence describe how a student experienced participating in the team, rather than a change in what the team did together. We report these because they shaped students’ willingness to engage, but distinguish them from effects such as faster feedback cycles or reduced peer burden, which more directly altered team-level coordination.

4.2.2 Theme 3: AI Does Not Solve Core Team Problems

Despite reported benefits, students expressed core teamwork problems as fundamentally interpersonal rather than technical. Unequal participation, scheduling conflicts, communication difficulties, and interpersonal tensions persisted regardless of their AI use. Students did not describe AI as capable of addressing these issues as they *can’t even imagine that the machine can help solve the problems between people* (NA-08).

Communication issues stayed central to team difficulties. Students described situations in which team members avoided discussing problems, withdrew from conversations, or reacted defensively to criticism. Unequal workload distribution, inconsistent participation, and a lack of open communication created additional frustration and resentment within teams:

“It was challenging. Teammates do not communicate that a task is too hard, and then wait for others to ask them about it. It delays the sprint not because someone is incapable, but because the person feels shame.” – EU2-77

31:12 How Students Construct Human–AI Boundaries in Software Engineering Teamwork

Students described situations where team members were *ghosting without notice to the team* (EU2-76) or contributed significantly more than others. These imbalances are connected to *scheduling*, different levels of time availability and priorities. On top of that, *interpersonal conflict and feedback* created emotional challenges as students struggled with how team members received criticism as

“*constructive criticism was rather seen as a personal attack against someone and instead of accepting feedback and suggestions for change, some people got mad and defensive.*” – EU2-19

Some students, particularly women, also described experiences of exclusion and disrespect that affected their sense of belonging within the team environment:

Diversity-related conflict emerged for some students, particularly women. They reported feeling excluded or disrespected by their teammates, to an extent where a woman reported that their team mates *were so disrespectful, that they made me cry* (EU2-02). These experiences went beyond task-related disagreements to affect students’ sense of belonging and psychological safety:

“*I had the feeling that the others are not interesting, but we are a team. It was a horrible feeling, because what was happening is that, not to make it a gender thing, I was left out of quite a bit of conversations. At some point, I truly felt like the crazy lady.*” – EU1-01

RQ2. AI improved conditions *surrounding* collaboration, but did not address core interpersonal challenges within teamwork.

4.3 RQ3: Socio-technical Relationship

As a recurring pattern across our data, students maintained a clear distinction between AI systems and human collaborators (*Theme 4*, Table 1). While AI was accepted as cognitive and technical support, students reserved social, emotional, and accountable aspects of teamwork for humans (*Theme 5*). That an AI system lacks empathy might be an unsurprising starting point; what we report below is how students translate this into specific exclusions: which situations, tasks, and relationships they withhold from AI, and why.

4.3.1 Theme 4: Constructing a Human–AI Boundary

AI as tool, not teammate. Students explicitly positioned AI as a tool rather than as a collaborative partner. They accepted AI support for technical and cognitive tasks, while rejecting the idea that AI could meaningfully participate in interpersonal collaboration *because human brain is not a computer and what happens there is way too complicated for a language model to grasp* (EU1-05).

“*If AI wouldn’t help with code, it wouldn’t help with people, interaction, and human things. We call it artificial intelligence, but in reality, it’s still not intelligence. It is just a a very profound and cutting edge piece of software.*” – EU1-10

Students expressed distrust towards AI in sensitive contexts. They raised concerns about bias and the lack of contextual understanding in AI systems:

“We didn’t use it for anything indigenous-related in the design sprint, because there’s a lot of bias, and AI inherently has white supremacy built into it. I trust my team mates.. or at least they would not do any damage with this kind of information. I wouldn’t trust any silicon company to knowing properly about how to interact with indigenous people.” – NA-02

Recognising AI’s emotional limitations. Students emphasised that while AI can be useful and efficient, it cannot replace the emotional and social qualities of teamwork. They described human collaboration as more meaningful, motivating, and emotionally rich. Students stated the emotional aspect, as *talking to humans requires empathy* (EU2-29), which the AI lacks.

“AI couldn’t provide emotional support, resolve conflicts, or offer real empathy during stressful moments. For me personally, it is this shared stress management that made my team experience unique.” – EU2-34

Others contrasted the value of human discussion with AI support, emphasising that the social dimension of teamwork cannot be replicated:

“The support from the team, especially when you are stuck with a topic, is much more valuable. And working in person and discussing with each other and also having fun as a group is something that AI can’t replace.” – EU2-43

Students described that human interaction allows richer understanding because relationships, also in professional settings, matter:

“with my team partner i want to find a solution, because i have to work with them in the future or even like them as humans. this is complicated but also more rewarding.” – EU2-73

Reserving empathy and support for humans. Students strongly rejected the idea of using AI for emotional or interpersonal problems. They emphasised that empathy, trust, encouragement, and understanding belong to human relationships.

“I like to evaluate what’s going on with people on my own, like reading social clues. I would ask a real person, not a language model.” – EU1-07

Students especially emphasised the value of human feedback and mentorship. When facing difficulties, they turned to teammates or instructors rather than AI:

“I’d be more likely to just whenever [Team member] and I couldn’t get over like, a hurdle. We would bring in [Mentor]. I would never, like never ever, asked chatgpt. It’s not human, and I don’t trust it with this kind of stuff.” – NA-06

Students also valued the critical perspective that only humans can provide in order to get diverse perspectives:

“But of course AI cannot replace critical pushback from humans which is necessary to avoid being trapped in an echo chamber.” – EU2-15

Maintaining human responsibility and accountability. Students also expressed that responsibility should remain with humans. They worried about errors, misleading output, and overreliance, and emphasised that human judgement remains essential, and their trust in human partners. They did not feel comfortable relying on AI in ways that would shift accountability away from themselves or their team members:

“At the end of the day a developer is the pilot of the project, not the LLM.” – EU2-22

Even in contexts where students engaged with online communities, they trusted collective human intelligence on platforms like Stack Overflow more than AI-generated solutions:

“When I have a problem, I’m going always to Stack Overflow, or whatever, because when you find a solution there, it will be rated. People have tested it. People have responded. You can feel the feedback, feel if it’s working or not.” – EU1-10

4.3.2 Theme 5: Tensions Between Efficiency and Values

Students described emerging tensions within teams related to different levels of AI adoption and different attitudes towards *appropriate* use. These differences became visible in *poor final work* (EU2-44), created concerns about fairness, and revealed disagreements about learning versus productivity gains. This created challenges for technical aspects, e.g. affecting code review, coding style and coherence and making it difficult to maintain consistency across the team’s work:

“It’s not very good if you are doing homework with GPT, and it’s plagiarizing and it’s gonna make your coding style look very, very weird. You’re not gonna have your own coding style, because GPT is just gonna generate code based on whatever like it finds on internet.” – NA-05

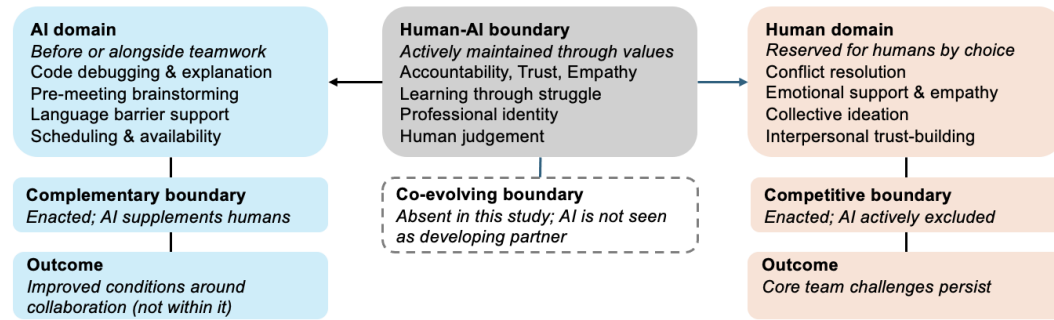
Rather than valuing efficiency alone, students associated learning with effort, discussion, and shared struggle, and worried that AI could weaken this process.

“I want to learn, right? With GPT, you get everything ready for you, you’re not really learning efficiently. When you learn new words, it’s better to see them in sentences and try to figure out the idea... Same with coding. If you get a piece of code that is written for you... you are not putting your effort into thinking of the logic behind this.” – EU1-09

They appreciated having discussions and disagreements, since they reported this is the key aspect of teamwork, exchanging perspectives and arguments. Students emphasised this benefit by noting that intentional non-use of AI led to more team interaction:

“The decision not to use any AI tools was a good one, it led to much more exchange and interaction within the team, even though it was getting a little bit heated, but we got really good ideas out of it.” – EU2-14

They associated meaningful learning with human interaction, especially when it comes to creativity, e.g. when asking for app names, the AI delivered *just kind of generated, random ones. It was like very cliché and gross. Once we had a proper sitdown with the team, we were super creative* (NA-10).



■ **Figure 2** Boundary work in student AI-human software teamwork based on Kim et al. [17] boundary types. Enacted boundaries shown solid, absent boundary dashed.

Overall, students accepted AI as support for individual cognition and preparation while maintaining clear boundaries around human responsibility, emotional engagement, and collaborative legitimacy. In their explanations, real discourse and emotional engagement only happen between humans:

“In the team, genuine discourse takes place, where ideas are developed and questioned together. In my opinion, emotional aspects such as motivation, mutual support, and creative engagement arise exclusively through teamwork, not through AI.” – EU2-39

RQ3. Students valued AI for cognitive support while reserving socially meaningful collaboration for humans.

5 Discussion

Our findings reveal a dichotomy of the perception of AI-human relationships: students across our cases described adopting AI tools (*RQ1*), and reported tangible benefits (*RQ2*), yet actively resist treating AI as part of their collaborations (*RQ3*). We frame these findings within boundary work and discuss their implications for software engineering.

5.1 AI Supports Collaboration Indirectly

The central contribution of this study is the distinction between AI supporting *around* collaboration versus *within* collaboration. Our contribution lies not in showing that students treat AI as a tool, which might be an expected outcome given current AI capabilities, but in specifying where and why they draw the line: which tasks they hand to AI, and which they actively withhold and why. As a recurring pattern across our data, students enacted complementary boundaries [17] by using AI for drafting, debugging, organising, brainstorming, and communication support *before* entering collaborations (Figure 2). Students described AI as functioning as preparatory infrastructure that strengthened individual readiness and reduced coordination friction surrounding teamwork.

At the same time, students draw strong competitive boundaries around empathy [8], responsibility [43], learning, and trust [36], rejecting AI for conflict resolution, emotional support, or collective accountability. AI benefits occur *before* or *alongside* teamwork, not *during* it. AI was therefore perceived as useful for supporting technical and cognitive work, but not for replacing socially accountable collaboration. These competitive boundaries were not uniform in kind. Some rested on accountability: students located responsibility for

outcomes in human judgement, treating AI participation as a threat to who could be held answerable for a decision (EU2-22) [43]. Others rested on trust: students questioned whether AI could be relied upon in contexts involving values, bias, or lived experience, preferring the tested judgement of human peers (NA-02, EU1-10). A third group of justifications was relational and emotional: students described qualities such as empathy, shared struggle, and motivation as constitutively human rather than technically absent from current AI (EU2-29, EU2-73) [8]. These grounds for exclusion are distinct and may not be equally durable: accountability-based boundaries could weaken as AI systems become more auditable and traceable, whereas relational boundaries are tied to what students consider collaboration to mean and may prove more resistant to changes in AI capability alone.

We found no evidence in students' accounts that they perceived AI as a co-evolving collaborative partner [17] (Figure 2). Kim et al. [17] studied individual use of generative AI (e.g. as a writing aid) and found that co-evolving boundaries can emerge over time. Our team setting did not reproduce this: unlike individual use, teamwork carries collective accountability and interpersonal stakes, which may make a co-evolving, partner-like framing of AI harder to sustain. Our students framed AI as a static support tool. Our findings therefore suggest that, within student software teams, AI is currently positioned more as infrastructural support than as a socially legitimate collaborator (Figure 2). As our design relies on retrospective interviews and reflections rather than longitudinal observation, this finding describes an absence in students' accounts at one point in time rather than a claim that students' boundaries cannot evolve at all.

This positioning may also explain why students with professional experience appeared more sceptical towards AI-generated output. These students often emphasised accountability, verification, and human judgement, reflecting concerns already discussed in professional human-AI collaboration research [43, 36].

However, our findings sit against a backdrop in which AI-as-teammate is so far mostly a projected future [1] or an individual-level, subordinate relationship [48] rather than an established discourse of equal collaboration, as human–machine teaming itself remains rare in software engineering practice [33]. Yet, our findings might be not applicable for explicitly pair programming, where the developers only have one partner, and thus might be less critical of an AI-partner [44]. As a research community, we should foster discussions on value conflicts and professional identity, contributing to the emerging debate on the role of software engineers in the AI era [3].

Future software engineers require not only technical AI literacy, but also opportunities to critically reflect on responsibility, trust, and acceptable forms of AI participation within collaborative software engineering work.

5.2 Valuing Human Collaboration Despite Its Difficulties

Students uphold human responsibility because they value empathy, critical judgment, and contextual understanding: competencies central to professional software engineering [10]. In this respect, the students appear well-prepared for professional practice. They also explicitly value the human character of collaboration and learning, even while acknowledging the difficulties they experience in teamwork.

Importantly, core teamwork problems persisted regardless of AI use. Unequal participation, missed deadlines, weak communication, defensive responses to feedback, and interpersonal tensions remained unchanged. Students neither described AI as helping with these challenges nor expected it to. These findings reinforce prior work suggesting that collaboration challenges

in software engineering are primarily social rather than technical [12, 20]. While AI may reduce organisational friction and support individual productivity, it does not replace the interpersonal competencies required for collaborative software development.

Students therefore appeared to maintain human responsibility not despite teamwork difficulties, but partly because collaboration itself was associated with learning, accountability, and shared ownership. If, as recent work suggests, AI offers productivity gains but cannot address the root causes of collaboration issues, then research, education, and practice must value human collaborative competencies and emerging tools *equally* [1].

Software engineering education and practices should even more emphasise communication, negotiation, constructive feedback, and conflict management.

5.3 AI, Inclusion, and Emerging Team Norms

Our findings further suggest that AI can both support and complicate inclusion within software teams. Some students, particularly international students, described AI as helpful for overcoming language barriers and participating more confidently in discussions. In these cases, AI reduced communicative friction and supported participation. This aligns with research on AI-supported task communication [15] and *technology as mediator* [27]. Importantly, students still distinguished this preparatory support from what they considered *real* collaboration.

At the same time, AI introduced new tensions. Prior work shows that unequal access to technology can exclude students [27]. Our findings extend this by showing that differences in beliefs about *appropriate* AI use and learning values may result in low (code) quality, and identity also shapes inclusion [11], and commitment to the team [45]. Our data suggest a possible explanation: students value disagreement and shared struggle. When AI removes these experiences, it may weaken ownership and connection [43].

These findings suggest that AI adoption within teams is not solely a technical issue, but also a sociotechnical negotiation of norms, values, and responsibility. Students actively constructed boundaries around acceptable AI participation while simultaneously negotiating what meaningful collaboration should look like in AI-supported software engineering work.

Within courses, educators could make those boundary-setting explicit, for example through structured team charters (as some of our participating teams already used informally) that specify where AI is and is not used, paired with rubrics that credit visible communication, conflict resolution, and negotiation rather than only technical output.

Future work should explore how cultural and professional context as well as exclusion experiences shape students' human-AI boundaries.

6 Conclusions

Our study shows how students draw boundaries around AI when working together in software engineering courses. Building on boundary work theory, we extend its application from individual AI use to teamwork and identify an important distinction between using AI to support collaboration and allowing AI to participate in it. Our findings indicate that students maintain clear boundaries between AI tools and human collaborators in project-based software courses. They mainly use AI for individual work and preparation, which can then support collaboration, for example by helping them communicate ideas more clearly. At the same time, common teamwork challenges persist regardless of AI use, and

students make deliberate decisions about where AI should and should not be involved. These boundaries help preserve what students see as the human core of teamwork, while also limiting AI's role in collaboration. Future research should examine how these boundaries vary across collaboration settings, cultural contexts, domains, tasks, and levels of experience, including professional practice. Longitudinal and multi-cohort studies could further show how accountability-, trust-, and relational boundaries change as AI capabilities and social norms evolve. For software engineering education, our findings suggest that AI literacy should distinguish between boundaries that may reasonably shift as AI improves and those that reflect more fundamental views about what collaboration should involve.

Data Availability Statement

Due to GDPR and participant confidentiality requirements, the original interview transcripts cannot be made publicly available. The full codebook, including examples, and the complete interview and reflection guides are available at: <https://figshare.com/s/578a68feb95cc8698de1>.

References

- 1 Silvia Abrahão, John Grundy, Mauro Pezzè, Margaret-Anne Storey, and Damian Tamburri. Software Engineering by and for Humans in an AI Era. *ACM Transactions on Software Engineering and Methodology*, January 2025.
- 2 Christiane Attig, Luisa Winzer, Tim Schrills, Maged Mortaga, Patricia Wollstadt, Christiane Wiebel-Herboth, and Thomas Franke. Understanding successful human–ai teaming: The role of goal alignment and ai autonomy for social perception of llm-based chatbots. *Computers in Human Behavior Reports*, 8:100246, 2025. doi:10.1016/j.chbah.2025.100246.
- 3 Stefanie Betz and Birgit Penzenstadler. With Great Power Comes Great Responsibility: The Role of Software Engineers. *ACM Transactions on Software Engineering and Methodology*, 34(5):1–21, June 2025. doi:10.1145/3715112.
- 4 Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.
- 5 Virginia Braun and Victoria Clarke. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health*, 11(4):589–597, 2019.
- 6 Virginia Braun and Victoria Clarke. One size fits all? what counts as quality practice in (reflexive) thematic analysis? *Qualitative research in psychology*, 18(3):328–352, 2021.
- 7 Virginia Braun and Victoria Clarke. Conceptual and design thinking for thematic analysis. *Qualitative psychology*, 9(1):3, 2022.
- 8 Lidiany Cerqueira, João Pedro Bastos, Danilo Neves, Glauco Carneiro, Rodrigo Spinola, Sávio Freire, Jose Santos, and Manoel Mendonça. Exploring Empathy in Software Engineering: Insights from a Grey Literature Analysis of Practitioners' Perspectives. *ACM Trans. Softw. Eng. Methodol.*, July 2025.
- 9 Na Chen and Xinyue Zhang. When misunderstanding meets artificial intelligence: the critical role of trust in human–ai and human–human team communication and performance. *Frontiers in Psychology*, 16:1637339, 2025.
- 10 Matheus De Morais Leça and Ronnie De Souza Santos. Curious, Critical Thinker, Empathetic, and Ethically Responsible: Essential Soft Skills for Data Scientists in Software Engineering. In *47th International Conference on Software Engineering: Software Engineering in Society*, 2025.
- 11 Shayna Earle, Madison McDonald, Esra Bengizi, and Kim S. Jones. Will I fit? The impact of social and identity determinants on teamwork in engineering education. In *Frontiers in Education*, volume 9, page 1412882. Frontiers Media SA, 2024.

- 12 Isabella Graßl, Stephan Krusche, and Gordon Fraser. Diversity and Teamwork in Student Software Teams. In *Proceedings of the 5th European Conference on Software Engineering Education*, pages 110–119, Seon/Bavaria Germany, June 2023. ACM. doi:10.1145/3593663.3593687.
- 13 Meryem Gunsaldi, Elif Guner, Musa Uckan, and Kaan Bati. The Impact of Generative AI Applications on Student Learning Outcomes in Science Education: A Systematic Review. *Journal of Education in Science, Environment and Health*, 11:196–208, June 2025.
- 14 Valerie J. Janesick. Peer Debriefing. In George Ritzer, editor, *The Blackwell Encyclopedia of Sociology*. Wiley, 1 edition, October 2015.
- 15 Harang Ju and Sinan Aral. Collaborating with ai agents: A field experiment on teamwork and communication patterns. <http://dx.doi.org/10.2139/ssrn.5187466>, 2025.
- 16 Natalie Kiesler, Jacqueline Smith, Juho Leinonen, Armando Fox, Stephen MacNeil, and Petri Ihantola. The Role of Generative AI in Software Student CollaborAtion, 2025. doi:10.48550/arXiv.2501.14084.
- 17 Young Ji Kim, Ceciley Xinyi Zhang, and Chengyu Fang. Navigating the human-ai divide: Boundary work in the age of generative ai. *Computers in Human Behavior: Artificial Humans*, 2025. doi:10.1016/j.chbah.2025.100214.
- 18 Diana Kirk, Andrew Luxton-Reilly, and Ewan Tempero. Risks in Student Projects. In *Proceedings of the 24th Australasian Computing Education Conference*, pages 143–152, Virtual Event Australia, February 2022. ACM. doi:10.1145/3511861.3511877.
- 19 Vassilka D. Kirova, Cyril S. Ku, Joseph R. Laracy, and Thomas J. Marlowe. Software Engineering Education Must Adapt and Evolve for an LLM Environment. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pages 666–672, Portland OR USA, March 2024. ACM. doi:10.1145/3626252.3630927.
- 20 Kamilla Kopeck-Harding, Sukru Eraslan, Bowen Cai, Suzanne M. Embury, and Caroline Jay. The impact of unequal contributions in student software engineering team projects. *Journal of Systems and Software*, 206:111839, December 2023. doi:10.1016/j.jss.2023.111839.
- 21 Ann Langley, Kajsa Lindberg, Bjørn Erik Mørk, Davide Nicolini, Elena Raviola, and Lars Walter. Boundary work among groups, occupations, and organizations: From cartography to process. *Academy of management annals*, 13(2):704–736, 2019.
- 22 Per Lenberg, Robert Feldt, Lucas Gren, Lars Göran Wallgren Tengberg, Inga Tidefors, and Daniel Graziotin. Qualitative software engineering research: Reflections and guidelines. *Journal of Software: Evolution and Process*, 36(6):e2607, 2024. doi:10.1002/smr.2607.
- 23 Yunbo Lyu, Zhou Yang, Jieke Shi, Jianming Chang, Yue Liu, and David Lo. "My Productivity Is Boosted, but ..." *Demystifying Users' Perception on AI Coding Assistants*. ASE, August 2025.
- 24 Courtney Miller, Rudrajit Choudhuri, Mara Ulloa, Sankeerti Haniyur, Robert DeLine, Margaret-Anne Storey, Emerson Murphy-Hill, Christian Bird, and Jenna L. Butler. "Maybe We Need Some More Examples:" Individual and Team Drivers of Developer GenAI Tool Use, July 2025. doi:10.48550/arXiv.2507.21280.
- 25 Nikolaos Mittas, Dimitrios Trygoniaris, Apostolos Ampatzoglou, Elvira-Maria Arvanitou, Christina Volioti, Alexander Chatzigeorgiou, and Lefteris Angelis. Required knowledge, skills and transversal competences for a career in software engineering. *Information and Software Technology*, 190:107949, February 2026. doi:10.1016/j.infsof.2025.107949.
- 26 Morteza Moalagh, Syed Sajid Hussain, Babak A. Farshchian, and Samrawit G. Selassie. Which teamwork challenges do computing students face in a project-based learning course in research methods? In *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*, pages 1–12, Koli Finland, November 2024. ACM. doi:10.1145/3699538.3699563.
- 27 Rantheke Jones Moloisane and Nicky Wolmarans. Bridging Gaps in Engineering Teamwork: A Literature Review through the Lens of Cultural-Historical Activity Theory. In *Proceedings from the International Research Symposium on Problem-Based Learning.*, 2025.

- 28 Helen Noble and Roberta Heale. Triangulation in research, with examples. *Evidence Based Nursing*, 22(3):67–68, July 2019.
- 29 James Prather, Juho Leinonen, Natalie Kiesler, Jamie Gorson Benario, Sam Lau, Stephen MacNeil, Narges Norouzi, Simone Opel, Vee Pettit, Leo Porter, Brent N. Reeves, Jaromir Savelka, David H. Smith, Sven Strickroth, and Daniel Zingaro. Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education*, pages 300–338, Milan Italy, January 2025. ACM. doi:10.1145/3689187.3709614.
- 30 Kai Presler-Marshall, Sarah Heckman, and Kathryn T. Stolee. What Makes Team[s] Work? A Study of Team Characteristics in Software Engineering Projects. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, pages 177–188, Lugano and Virtual Event Switzerland, August 2022. ACM. doi:10.1145/3501385.3543980.
- 31 Yinuo Qin, Richard T. Lee, and Paul Sajda. Perception of an ai teammate in an embodied control task affects team performance. *arXiv preprint*, 2025. arXiv:2501.15332.
- 32 Christian Rahe and Walid Maalej. How Do Programming Students Use Generative AI? *Proceedings of the ACM on Software Engineering*, 2:978–1000, June 2025. doi:10.1145/3715762.
- 33 Irum Rauf, Helen Sharp, Tamara Lopez, and Michel Wermelinger. Human-machine teaming and team effectiveness in ai tools for software engineering. In *Proceedings of the 2025 International Conference on Cooperative and Human Aspects of Software Engineering*. IEEE, 2025. doi:10.1109/CHASE66643.2025.00017.
- 34 Herbert Rubin and Irene Rubin. *Qualitative interviewing: The art of hearing data*. sage, 2011.
- 35 Zhang Sai, Zhenchang Xing, Ronghui Guo, Fangzhou Xu, Lei Chen, Zhaoyuan Zhang, Zhang Xiaowang, Zhiyong Feng, and Zhiqiang Zhuang. Empowering Agile-Based Generative Software Development through Human-AI Teamwork. *ACM Transactions on Software Engineering and Methodology*, 34, January 2025. doi:10.1145/3702987.
- 36 Anshul Shah, Thomas Rexin, Elena Tomson, William G Griswold, Leo Porter, and Adalbert Gerald Soosai Raj. Evolution of Programmers' Trust in Generative AI Programming Assistants. In *Proceedings of the 25th Koli Calling International Conference on Computing Education Research*, Koli Calling '25, pages 1–11, New York, NY, USA, November 2025. Association for Computing Machinery. doi:10.1145/3769994.3770029.
- 37 Margaret-Anne Storey, Neil A. Ernst, Courtney Williams, and Eirini Kalliamvakou. The who, what, how of software engineering research: A socio-technical framework. *Empirical Software Engineering*, 25(5):4097–4129, September 2020. doi:10.1007/s10664-020-09858-z.
- 38 Katarzyna Szmyd and Ewelina Mitera. The impact of artificial intelligence on the development of critical thinking skills in students. *European Research Studies Journal*, 27, 2024.
- 39 Claire Taylor, Marie Huber, Qiao Ma, Rayven Plaza, Alison Chang, and Jie Chen. Software Development Is a Team Sport. *IEEE Software*, 42(03):13–17, May 2025. doi:10.1109/MS.2025.3539403.
- 40 Valerio Terragni, Annie Vella, Partha Roop, and Kelly Blincoe. The Future of AI-Driven Software Engineering. *ACM Transactions on Software Engineering and Methodology*, page 3715003, January 2025.
- 41 Allison Tong, Peter Sainsbury, and Jonathan Craig. Consolidated criteria for reporting qualitative research (coreq). *International journal for quality in health care*, 19(6):349–357, 2007.
- 42 Christoph Treude and Margaret-Anne Storey. Generative AI and Empirical Software Engineering: A Paradigm Shift, February 2025. doi:10.48550/arXiv.2502.08108.
- 43 Mara Ulloa, Jenna L. Butler, Sankeerti Haniyur, Courtney Miller, Barrett Amos, Advait Sarkar, and Margaret-Anne Storey. Product Manager Practices for Delegating Work to Generative AI: "Accountability must not be delegated to non-human actors", October 2025. doi:10.48550/arXiv.2510.02504.

- 44 Alisa Welter, Niklas Schneider, Tobias Dick, Kallistos Weis, Christof Tinnes, Marvin Wyrich, and Sven Apel. From developer pairs to ai copilots: A comparative study on knowledge transfer. *arXiv preprint*, 2025. doi:10.48550/arXiv.2506.04785.
- 45 Suqing Wu, Yukun Liu, Mengqi Ruan, Siyu Chen, and Xiao-Yun Xie. Human-generative ai collaboration enhances task performance but undermines human’s intrinsic motivation. *Scientific Reports*, 15(1):15105, 2025.
- 46 Robert K. Yin. *Case Study Research: Design and Methods*, volume 5. sage, 2009.
- 47 Ilya Zakharov, Ekaterina Koshchenko, and Agnia Sergeyuk. AI in Software Engineering: Perceived Roles and Their Impact on Adoption, April 2025. doi:10.48550/arXiv.2504.20329.
- 48 Ilya Zakharov, Ekaterina Koshchenko, and Agnia Sergeyuk. From Teacher to Colleague: How Coding Experience Shapes Developer Perceptions of AI Tools, April 2025. doi:10.48550/arXiv.2504.13903.
- 49 Josie Zvelebilova, Saiph Savage, and Christoph Riedl. Collective attention in human-ai teams. *openreview.net*, 2024.